

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A method of debugging software comprising:
 - obtaining a software module;
 - obtaining a first input test vector;
 - obtaining a bug list, wherein said bug list comprises a minimal set of stimuli to reproduce a software bug in said software module;
 - generating a first output vector by applying said first input test vector to said software module;
 - applying a comparison test to said first output vector to determine whether a bug exists in said software module;
 - applying a module decomposition test to said software module when the result of said comparison test is positive; ~~and~~
 - obtaining a module decomposition list comprising two or more submodules of said software module and iteratively processing said module decomposition list when the result of said module decomposition test is positive, wherein iteratively processing said module decomposition list comprises removing at least one submodule; and
 - appending said software module and said first input test vector to said bug list when said software module is a minimal module based on module decomposition list as determined by the ~~result of said module decomposition test is negative.~~
2. (Original) The method of claim 1 wherein said comparison test comprises:
 - obtaining an optimal result vector;
 - comparing said first output vector to said optimal result vector; and
 - determining whether said first output vector is at variance with said optimal result vector.
3. (Canceled)
4. (Currently Amended) The method of claim ~~[[3]]~~ 1 wherein the iterative processing step comprises:

obtaining a second input test vector such that the application of said second input test vector to said two or more submodules ~~will~~ generates a second output test vector; and
recursively processing said two or more submodules and said second output test vector.

5. (Currently Amended) The method of claim 4 wherein the ~~trimming~~ recursively processing step comprises:

obtaining said minimal module;
obtaining said first input test vector; and
applying a vector decomposition test to said first input test vector.

6. (Original) The method of claim 5 further comprising:

generating a third output vector by applying said first input test vector to said minimal module when the result of said vector decomposition test is negative;
applying said comparison test to said third output vector to determine whether said first input test vector recreates the bug; and
appending said input test vector to a test list when the result of said comparison test is positive.

7. (Original) The method of claim 6 further comprising:

obtaining a vector decomposition list comprising two or more subvectors of said first input test vector when the result of said vector decomposition test is positive; and
iteratively processing each entry in said vector decomposition list by recursively applying said vector decomposition test to said subvectors.

8. (Original) The method of claim 7 in which said software module and said input test vector are obtained by iterating through the entries in said bug list.

9. (New) A computer system for debugging software comprising:

a processor;
a memory;
a storage device; and

software instructions stored in the memory for enabling the computer system under control of the processor, to:

- obtain a software module;
- obtain a first input test vector;
- obtain a bug list, wherein said bug list comprises a minimal set of stimuli to reproduce a software bug in said software module;
- generate a first output vector by applying said first input test vector to said software module;
- apply a comparison test to said first output vector to determine whether a bug exists in said software module;
- apply a module decomposition test to said software module when the result of said comparison test is positive;
- obtain a module decomposition list comprising two or more submodules of said software module and iteratively processing said module decomposition list when the result of said module decomposition test is positive, wherein iteratively processing said module decomposition list comprises removing at least one submodule; and
- append said software module and said first input test vector to said bug list when said software module is a minimal module based on module decomposition list as determined by the module decomposition test.